Dear reader,

What follows is a detailed description of one of the puzzles Mike Fellows and I have designed: *The Big Meanies of Corner Town*. I believe it's a solid logic puzzle of *Minesweeper's* order—but with more ramping possibilities and more story. The suggested story has warmth and humour together with basic conflict. The ramping possibilities make the puzzle suitable for ages 4 through adult. The ramping suggestions are dispersed throughout the document.

The educational value of the puzzle/game is twofold: it is a challenging exercise in logic at the higher levels of game play; also, it's a pleasant introduction to *graph theory* and the ideas of an *independent set of vertices* (a set of vertices that are not connected) and a *maximal independent set of vertices* (an independent set of vertices that cannot be added to and still be independent).

Mike Fellows has developed a method of generating such puzzles from the mathematics of theoretical Computer Science.

Warm regards,
Jim Andrews
Jim@vispo.com
http://www.vispo.com

The Big Meanies of Corner Town (I)

A logic puzzle/story/game for Broderbund
Age group: 4-adult (ramped)

From:

Jim Andrews
1806 8th Ave, #313
Seattle WA, 98101
USA

Jim@vispo.com
http://www.vispo.com

Dr. Michael Fellows
Dept. of Computer Science
P.O. Box 3025
University of Victoria
Victoria, B.C.
Canada, V8W 3P2

mfellows@csr.uvic.ca

THE BIG MEANIES OF CORNER TOWN (I)
(Discreet Repair of Maximal Independent Set)

The Story and Rules

You are presented with a map of a neighbourhood. It's a bit odd in Corner Town because all the houses are located at street corners and there's only one house per street corner. You see the streets and the houses, but you can't see who lives in the houses. But you do know that each house contains either a Big Meanie or a Peaceful person. You know that *Big Meanies fight incessantly if they live on the same street.* This neighbourhood is quite peaceable—except for *two* Big Meanies who live on the same street. There are probably more Big Meanies in the neighbourhood—but only *two* of them live on the same street.

The only way to solve the problem of the two Big Meanies living on the same street is to get rid of the street between them. They won't budge. *Get rid of that street and you win.* But be careful, because you can only get rid of *one* street (though easier versions of the game could allow you to get rid of more than one street). *If you get rid of the wrong street you lose*, game over, that's it.[1]

You also know that no Peaceful Person has it perfect in this neighbourhood: *every Peaceful Person lives on the same street as at least one Big Meanie.*

When you click on a house, the inhabitant is revealed. A very simple animation sequence might reveal a green and growling Big Meanie sticking out his tongue at you or perhaps he snorts or farts or whatever. He/she's plainly a Big Meanie. On the other hand, you may reveal a Peaceful person smiling at you or doing/saying something pleasant. Several of the pleasant people might mention the problem of the Big Meanies living on the same street somewhere in the neighbourhood and could you kindly nuke the road between them, please? These sorts of hints could be for beginners and could be turned off when the player becomes proficient at the game. Or their variety and wit, colour, and humour could be an attractive part of the game to enchant players. Part of the charm of the game will be in the variety and wit of the characters and what they say and look like. The comments could also explain briefly the above. "*None of us Peaceful people has it perfect here, you know: every Peaceful Person lives on the same street as at least one Big Meanie.*"

If you ever reveal both of the Big Meanies who live on the same street, you lose. The Big Meanies tangle big time and destroy the whole neighbourhood.

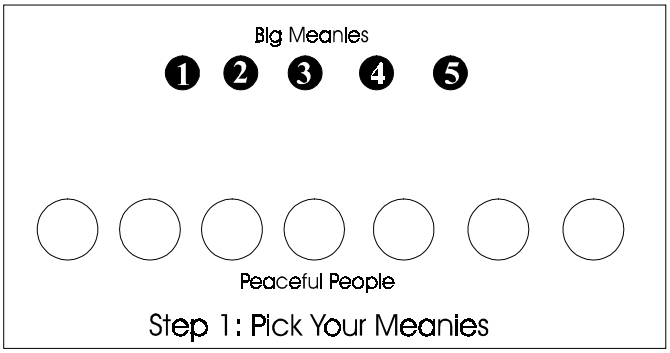
There are only two things a player can do: reveal an inhabitant or get rid of a street. And you can only get rid of a street once.

GENERATING AN INSTANCE

How does the program generate an instance of Corner Town in such a way that the player will hardly ever play the same game twice? Also, how do we ensure that every Peaceful Person lives on the same street as at least one Meany and only two Meanies live on the same street?

Step 1: Determine the Number of Houses in Corner Town and Put an Inhabitant in Each House

Begin with n vertices (houses). The actual value of n will depend on the ramping position of the player within the game. If Corner Town fits entirely on one screen, then n should be approximately 7 to 35. If Corner Town does not fit on one screen, n could be higher.[2]

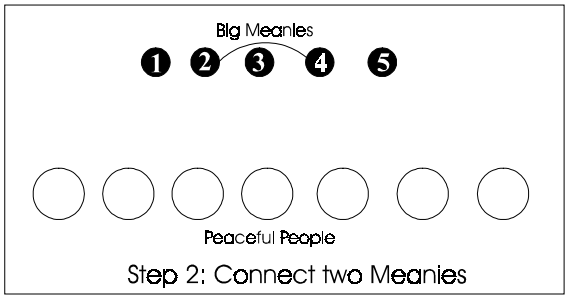


Next, choose the houses that contain Big Meanies. This can be done by visiting each house and making a random assignment of Big Meany or Peaceful Person to the house. One of the ramping parameters could be the probability value used here. For instance, the program may visit each house and, in turn, assign it a Big Meany with probability 0.6.[3] A prototype will have to include this parameter in the Options menu—the ratio of Big Meanies to Peaceful People will be important to the playability, challenge, and fun level of the game. The idea is to construct Corner Town in such a way that the player is required to guess as little as possible. But at higher levels the number of non-risky moves a player can make should be few as the game progresses.[4]

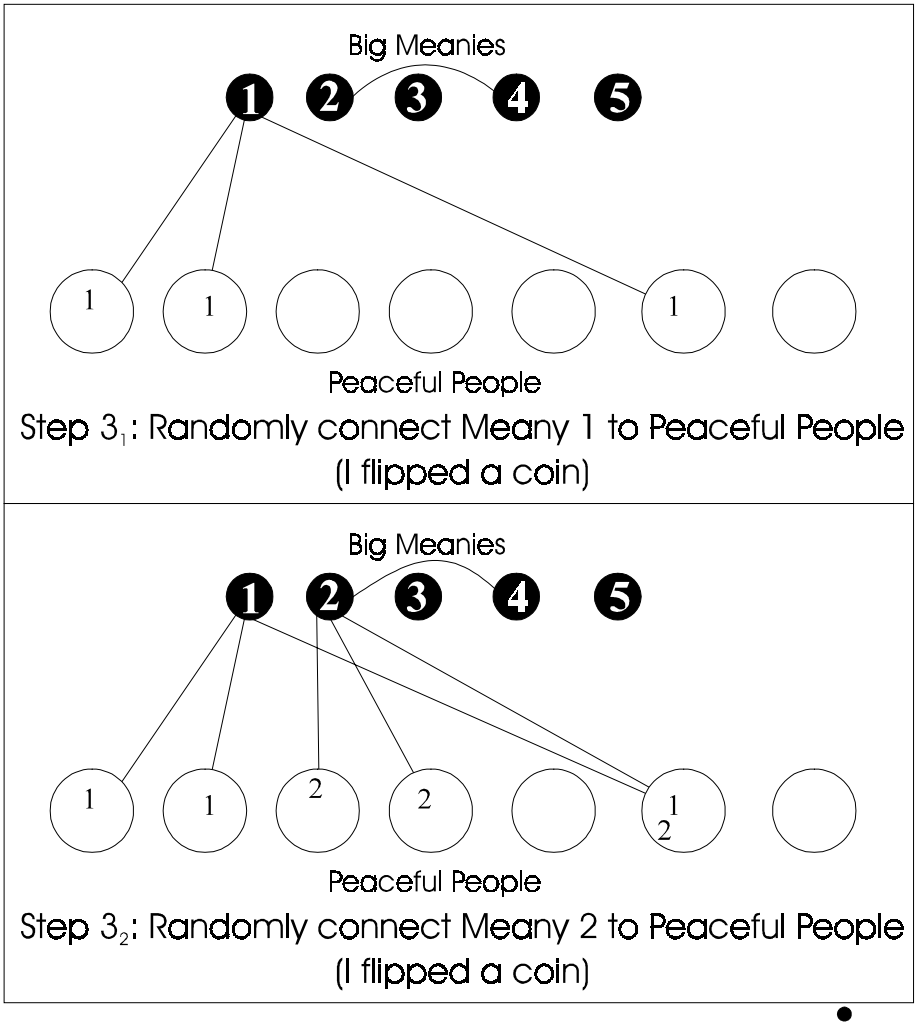
Initial investigation suggests that when you construct sample games to play out and understand the game more deeply, you should use a ratio of 50% or less of Meanies. But a prototype is required to test the situation further to determine the ramping status of this parameter.

Step 2: Put Two (And Only Two) Big Meanies on the Same Street

Randomly connect the houses of two Big Meanies. This assignment of a street connecting the houses of two Big Meanies is essential. *Furthermore, never connect any more than one pair of Big Meanies during the construction of the town.*[5] This property of the town allows the player to make certain inferences.

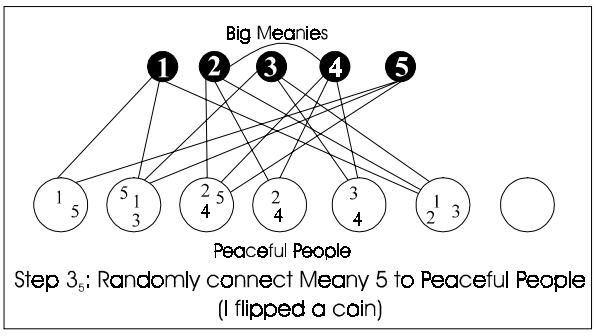


Step 3: Form Streets Between Big Meanies and Peaceful People



Start with Big Meany 1. Form a street between Big Meany 1 and Peaceful Person 1 with probability *y* (*y* is a number between 0 and 1). Then form a street between Big Meany 1 and Peaceful Person 2 with probability *y*. Do this for each Peaceful Person.

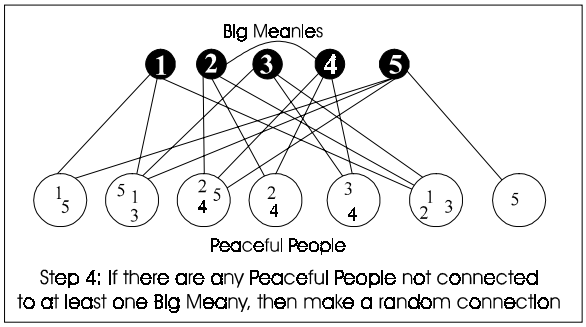
Then move on to form streets between Big Meany 2 and the Peaceful People with probability *y*. Do this for each Big Meany.



Step 4: If Any Peaceful People are Not Connected to At Least One Meany, Connect Them To A Random Meany.

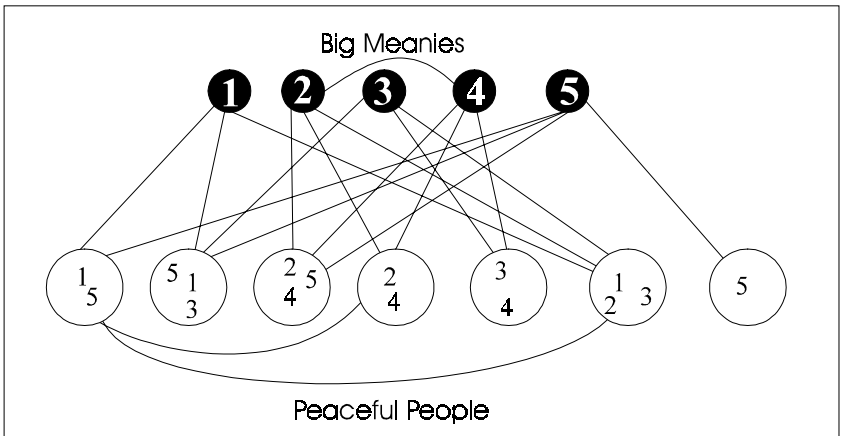
It is important to the game that each Peaceful Person live on the same street as at least one Big Meany (it allows the player to make certain inferences).

So inflict a Big Meany on any Peaceful Person who doesn't have a Meany on the same street.

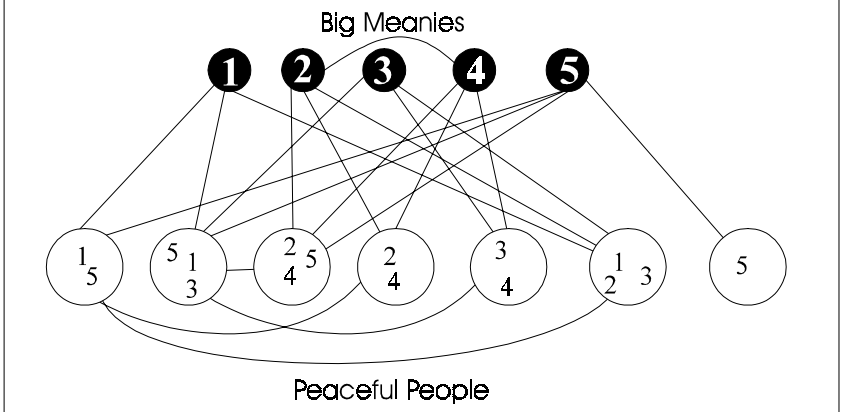


Step 4: If there are any Peaceful People not connected to at least one Big Meany, then make a random connection

Step 5: Connect Peaceful People to Peaceful People

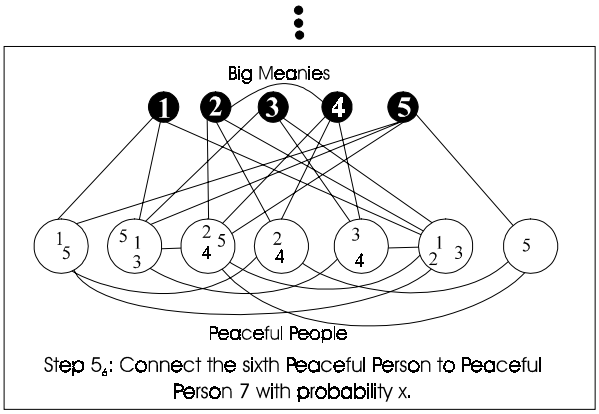


Step 5_1: Connect the first Peaceful Person to other Peaceful people with probability x (I used $x=0.\overline{3}$).
This also can be a ramping parameter (consider x=0 vs. x=1)



Step 5_2: Connect the second Peaceful Person to other Peaceful people 3-7 with probability x.

A prototype is required to determine whether the parameter x really influences the playability, fun, and challenge of the game or whether the value of x should simply be *y*.

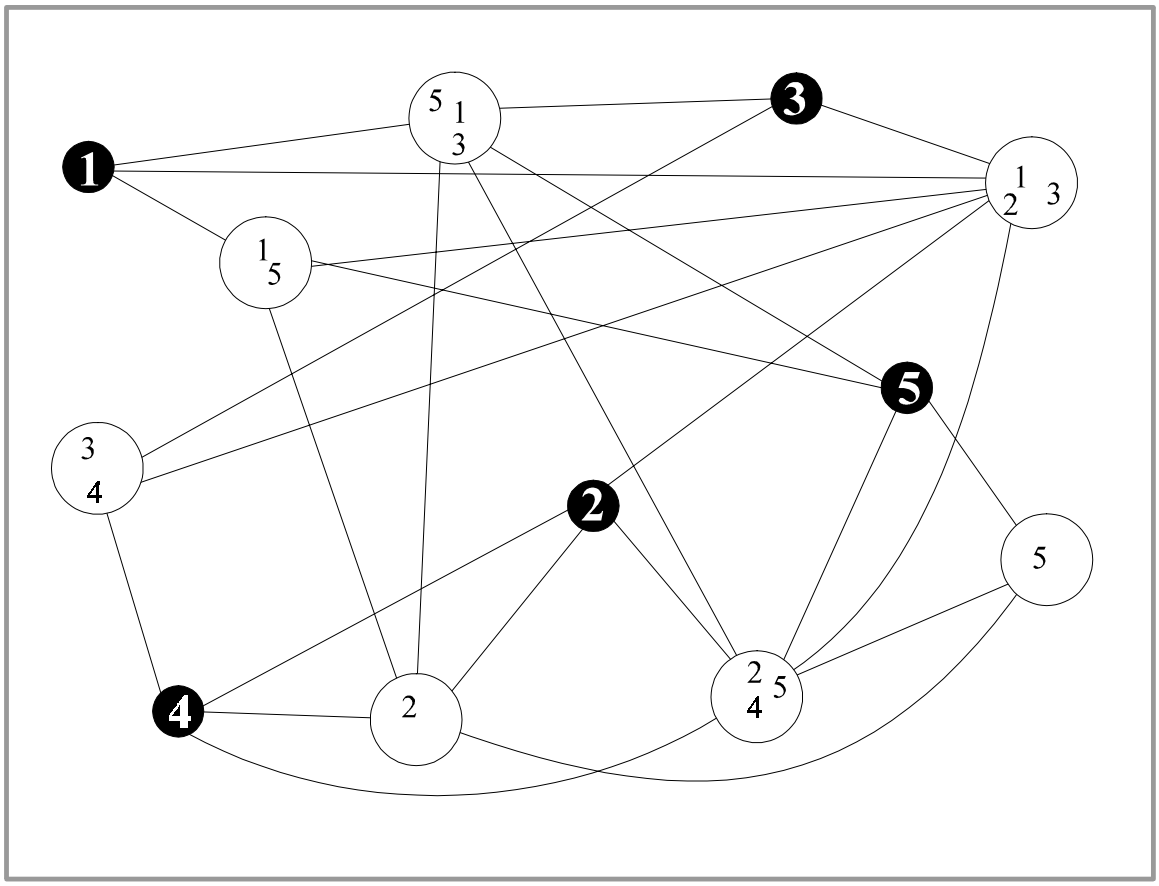


Step 5_6: Connect the sixth Peaceful Person to Peaceful Person 7 with probability x.

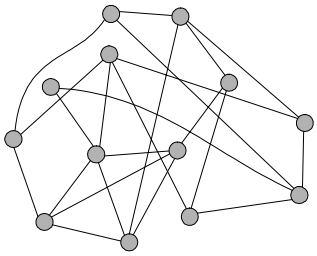
Remarks on the Construction

The game on the next page is actually the one above rearranged to look a bit prettier (though not much prettier). I played it through twice (covering the houses with coins) without having to make any risky guesses. Good strategies (like the one outlined in the diagrams of the game play-through) result in having to guess very little. However, bad strategies result in having to guess more than is healthy for the good of the neighbourhood. There might even be a theorem or two to prove about the maximum and minimum number of guesses required with different strategies. Is the minimum number of guesses always 0 when my strategy is used? Not likely, but I haven't been stymied yet. The values of the ramping parameters will undoubtedly affect the number of required guesses.

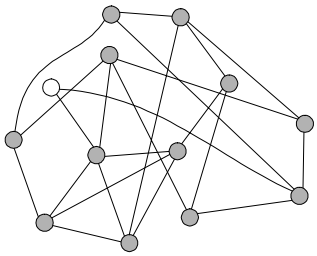
I didn't make any special attempts to make this particular game fancy. I just followed the recipe and didn't play it until I'd done the above diagrams and the diagram on the next page. In other words, the algorithm outlined for the construction of an enjoyable game seems like it might be close to tuned though a prototype that allows adjustment of the parameters I've mentioned will definitely be great for fine tuning and ramping.


1
3
5
1
3
1
2
3
1
5
5
3
4
2
5
4
2
2
4
5

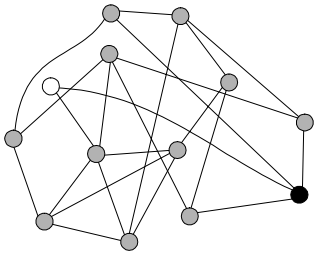
Sample Game Play-Through



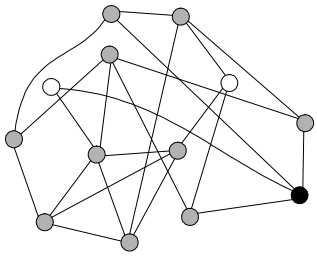
State 0: You can safely uncover any vertex



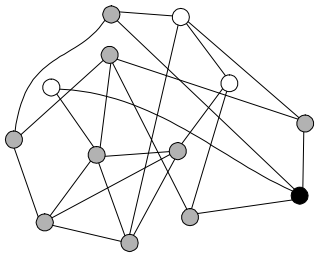
State 1: Your strategy will be to uncover vertices with few edges attached to them. That way, if the vertices turn out to be black, you've minimized the number of vertices you can't uncover.



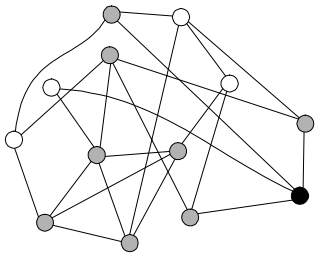
State 2: By uncovering the one you have, you have created 3 vertices you better not uncover. If you were now to uncover any of the vertices attached to the black one, you'd risk losing.



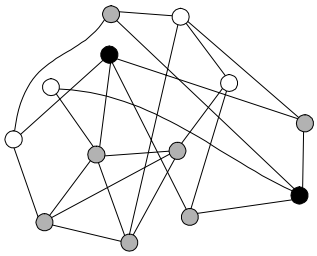
State 3: You uncover a safe one of degree 3



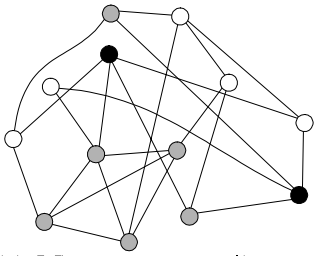
State 4: By uncovering this one, you create at most 2 new vertices that you can't uncover. But you got lucky: It's white, so you've created no problem vertices.



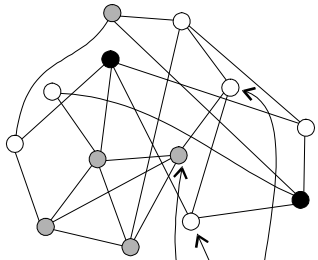
State 5: Again, by uncovering this one, you can create, at most, 2 problem vertices (it's attached to a problem vertex already).



State 6: By uncovering this one you create, at most, 1 new problem vertex (It's already attached to 2 problem vertices). AHA! Now you know two covered vertices must be white!

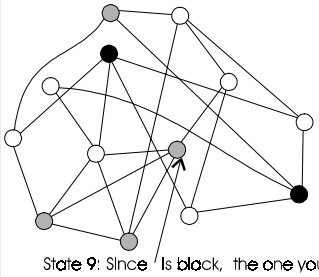


State 7: The one you uncovered here HAS TO BE WHITE: if it were black, then there would be 2 BAD EDGES. But there is only 1 bad edge.

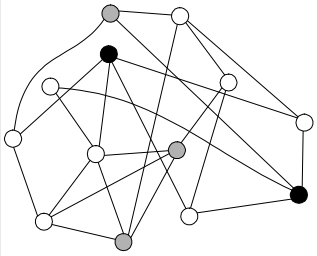


State 8: The one you uncovered must be white for the same reason as in the previous move.

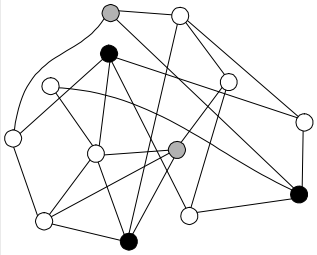
Now you can also infer that must be black. Why? Because if it were white then would be surrounded by only white vertices. But no white vertex on the graph is surrounded by only white vertices.



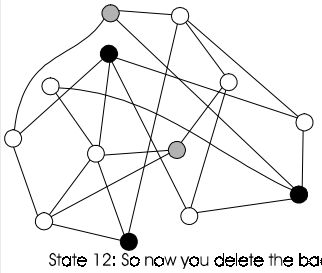
State 9: Since Is black, the one you've just uncovered must be white..O therwise there would be 2 bad edges.



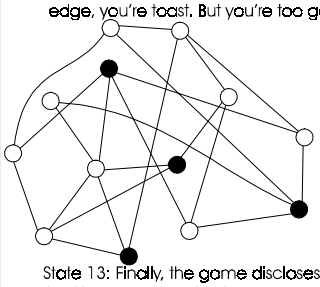
State 10: This one's still safe to uncover.



State 11: This one's still safe to uncover--and it's black! So now you know where the bad edge is--since you know that the bottom most covered vertex is actually black.



State 12: So now you delete the bad edge . You can only delete 1 edge during a game. If you delete the wrong edge, you're toast. But you're too good.



State 13: Finally, the game discloses itself to you, congratulates you on your fine play, and asks if you would like to play again.

Scoring

The player's score for the game should be inversely proportional to the amount of time the player took and directly proportional to the level of play. If the player wins, the formula could be something like Points awarded=(Level * 1000)/time. If the player loses, either no points are awarded or (to be more generous) the formula could be something like Points awarded= # of houses uncovered before the Meanies tangle.

Footnotes

[1] In easier levels, you lose if you run out of streets to get rid of without having solved the problem of the street between the two Big Meanies. This type of ramping would be important so that pre-school children could be introduced to the game. The serio-deductive nature of the logic puzzle would otherwise prove too challenging (wouldn't it?) for pre-schoolers. Although such ramping would also degrade the graph's structure (perhaps the resulting graph no longer has the special property that every Peaceful Person lives beside at least one Meany) the player would stand a good chance of winning. Another "ramping" parameter could be the size of the neighbourhood.

[2] Note that the diagrams do not reflect how the game will look on the screen. The diagrams divide the houses into clearly distinguishable sets of Meanies and Peaceful people to make the Instance Generation understandable.

[3] There has to be at least two Big Meanies in the town so that there can be two Big Meanies who live on the same street.

[4] Good strategies in *The Big Meanies of Corner Town* will lead to a situation where the player has to guess a minimum number of times whereas bad strategies will lead the player to guess several times. Notice that in the sample game play-through there is no risky guessing *at all*.

[5] The puzzle must be programmed in such a way that the streets are either straight lines (when no intermediary houses fall between the two houses on a street) or the streets are *elliptical arcs* (when one or more houses fall between the two houses on a street). It is important that the game not be developed simply on a grid that allows only for adjacent squares to be connected. Such graphs are limited in their appeal and interest.